Accredited

AS LEVEL
Specification

COMPUTER SCIENCE

H046
For first assessment in 2016

ocr.org.uk/alevelcomputerscience

OCR
Oxford Cambridge and RSA

We will inform centres about any changes to the specification. We will also publish changes on our website. The latest version of our specification will always be the one on our website **(ocr.org.uk)** and this may differ from printed versions.

Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee. Registered in England. Registered company number 3484466.

Registered office: 1 Hills Road
Cambridge
CB1 2EU.

OCR is an exempt charity.

Contents

Introducing…
AS Level Computer Science (from September 2015)

Computer Science is a practical subject where students can apply the academic principles learned in the classroom to real-world systems. It's an intensely creative subject that combines invention and excitement, and can look at the natural world through a digital prism.

The aims of this qualification are to enable learners to develop:

- An understanding and ability to apply the fundamental principles and concepts of computer science, including: abstraction, decomposition, logic, algorithms and data representation
- The ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs to do so
- The capacity to think creatively, innovatively, analytically, logically and critically
- The capacity to see relationships between different aspects of computer science
- Mathematical skills.

Meet the team

We have a dedicated team of people working on our AS Level Computer Science qualifications.

Find out more about our Computer Science team at ocr.org.uk/computerscienceteam

If you need specialist advice, guidance or support, get in touch as follows:

- 01223 553998
- computerscience@ocr.org.uk
- @OCR_ict

Teaching and learning resources

We recognise that the introduction of a new specification can bring challenges for implementation and teaching. Our aim is to help you at every stage and we're working hard to provide a practical package of support in close consultation with teachers and other experts, so we can help you to make the change.

Designed to support progression for all

Our resources are designed to provide you with a range of teaching activities and suggestions so you can select the best approach for your particular students. You are the experts on how your students learn and our aim is to support you in the best way we can.

We want to…

- Support you with a body of knowledge that grows throughout the lifetime of the specification
- Provide you with a range of suggestions so you can select the best activity, approach or context for your particular students
- Make it easier for you to explore and interact with our resource materials, in particular to develop your own schemes of work
- Create an ongoing conversation so we can develop materials that work for you.

Plenty of useful resources

You'll have four main types of subject-specific teaching and learning resources at your fingertips:

- Delivery Guides
- Transition Guides
- Topic Exploration Packs
- Lesson Elements.

Along with subject-specific resources, you'll also have access to a selection of generic resources that focus on skills development and professional guidance for teachers.

Skills Guides – we've produced a set of Skills Guides that are not specific to Computer Science, but each covers a topic that could be relevant to a range of qualifications – for example, communication, legislation and research. Download the guides at ocr.org.uk/skillsguides.

Active Results – a free online results analysis service to help you review the performance of individual students or your whole school. It provides access to detailed results data, enabling more comprehensive analysis of results in order to give you a more accurate measurement of the achievements of your centre and individual students. For more details refer to ocr.org.uk/activeresults.

Professional development

Take advantage of our improved Professional Development Programme, designed with you in mind. Whether you want to come to face-to-face events, look at our new digital training or search for training materials, you can find what you're looking for all in one place at the CPD Hub.

An introduction to the new specifications

We'll be running events to help you get to grips with our AS Level Computer Science qualification.

These events are designed to help prepare you for first teaching and to support your delivery at every stage.

Watch out for details at **cpdhub.ocr.org.uk**.

To receive the latest information about the training we'll be offering, please register for AS Level email updates at **ocr.org.uk/updates**.

1 Why choose an OCR AS Level in Computer Science?

1a. Why choose an OCR qualification?

Choose OCR and you've got the reassurance that you're working with one of the UK's leading exam boards. Our new AS Level in Computer Science course has been developed in consultation with teachers, employers and Higher Education to provide students with a qualification that's relevant to them and meets their needs.

We're part of the Cambridge Assessment Group, Europe's largest assessment agency and a department of the University of Cambridge. Cambridge Assessment plays a leading role in developing and delivering assessments throughout the world, operating in over 150 countries.

We work with a range of education providers, including schools, colleges, workplaces and other institutions in both the public and private sectors. Over 13,000 centres choose our A levels, GCSEs and vocational qualifications including Cambridge Nationals and Cambridge Technicals.

Our Specifications

We believe in developing specifications that help you bring the subject to life and inspire your students to achieve more.

We've created teacher-friendly specifications based on extensive research and engagement with the teaching community. They're designed to be straightforward and accessible so that you can tailor the delivery of the course to suit your needs. We aim to encourage learners to become responsible for their own learning, confident in discussing ideas, innovative and engaged.

We provide a range of support services designed to help you at every stage, from preparation through to the delivery of our specifications. This includes:

A wide range of high-quality creative resources including:

- Delivery Guides
- Transition Guides
- Topic Exploration Packs
- Lesson Elements
- …and much more.

- Access to subject specialists to support you through the transition and throughout the lifetimes of the specifications.
- CPD/Training for teachers to introduce the qualifications and prepare you for first teaching.
- Active Results – our free results analysis service to help you review the performance of individual students or whole schools.
- ExamCreator – our new online past papers service that enables you to build your own test papers from past OCR exam questions.

All AS level qualifications offered by OCR are accredited by Ofqual, the Regulator for qualifications offered in England. The accreditation number for OCR AS Level in Computer Science is QN: 601/5030/0.

1b. Why choose an OCR AS Level in Computer Science?

"At its heart lies the notion of computational thinking: a mode of thought that goes well beyond software and hardware, and that provides a framework within which to reason about systems and problems."
(Computer Science a Curriculum for Schools).

This specification has been developed by the team that created the first GCSE Computing qualification in the UK. The experience of the past three years of assessment has clearly demonstrated that OCR has the knowledge and skills to develop reliable and valid qualifications in this area of study.

OCR Computer Science will above all else be relevant to the modern and changing world of computing. It enables teachers to be able to tailor the qualification to meet the needs of their learners in their centre and has an open source ethos allowing any programming language that meets the needs of the course to be used.

Computer Science is a practical subject where learners can apply the academic principles learned in the classroom to real world systems. It is an intensely creative subject that combines invention and excitement, that can look at the natural world through a digital prism. OCR's Computer Science will value computational thinking, helping learners to develop the skills to solve problems, design systems and understand the power and limits of human and machine intelligence.

Aims and learning outcomes

The aims of this qualification are to enable learners to develop:

- an understanding of and ability to apply the fundamental principles and concepts of computer science including; abstraction, decomposition, logic, algorithms and data representation
- the ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs to do so
- the capacity for thinking creatively, innovatively, analytically, logically and critically
- the capacity to see relationships between different aspects of computer science
- mathematical skills
- the ability to articulate the individual (moral), social (ethical), legal and cultural opportunities and risks of digital technology.

1c. What are the key features of this specification?

The OCR AS Level in Computer Science will encourage learners to be inspired, motivated and challenged by following a broad, coherent, practical, satisfying and worthwhile course of study. It will provide insight into, and experience of how computer science works, stimulating learners' curiosity and encouraging them to engage with computer science in their everyday lives and to make informed choices about further study or career choices.

The key features of this specification encourage:

- emphasis on problem solving using computers
- emphasis on computer programming and algorithms
- emphasis on the mathematical skills used to express computational laws and processes, e.g. Boolean algebra/logic and algorithm comparison
- less emphasis on ICT.

1d. How do I find out more information?

If you are already using OCR specifications you can contact us at: **www.ocr.org.uk**

If you are not already a registered OCR centre then you can find out more information on the benefits of becoming one at: **www.ocr.org.uk**

If you are not yet an approved centre and would like to become one go to: **www.ocr.org.uk**

Find out more?

Ask a Subject Specialist:

Email: **computerscience@ocr.org.uk**

Customer Contact Centre: 01223 553998

Teacher support: **www.ocr.org.uk**

News: **www.ocr.org.uk**

2 The specification overview

2a. Overview of AS Level in Computer Science (H046)

Learners must take both components (01 and 02) to be awarded the OCR AS Level in Computer Science.

<table>
<tr><th>Content Overview</th><th colspan="2">Assessment Overview</th></tr>
<tr><td rowspan="2">• The characteristics of contemporary processors, input, output and storage devices
• Software and software development
• Programming
• Exchanging data
• Data types, data structures and algorithms
• Legal, moral, ethical and cultural issues

• Elements of computational thinking
• Problem solving and programming
• Algorithms</td><td>Computing principles (01)
70 marks
1 hour and 15 minutes
written paper</td><td>50%
of total
AS level</td></tr>
<tr><td>Algorithms and problem solving (02*)
70 marks
1 hour and 15 minutes
written paper</td><td>50%
of total
AS level</td></tr>
</table>

* Indicates synoptic assessment

2b. Content of AS Level in Computer Science (H046)

The content of this AS Level in Computer Science is divided into two components:

- Computing principles component (01) contains the majority of the content of the specification and is assessed in a written paper recalling knowledge and understanding.
- Algorithms and problem solving component (02) relates principally to problem solving skills needed by learners to apply the knowledge and understanding encountered in the Computing principles component.

This specification has been designed to be co-teachable with the stand alone A Level in Computer Science.

Mathematical skills are embedded throughout the content of the two components and assessed in the written papers where appropriate. The quality of extended responses are assessed in the written papers where indicated by an asterisk.

2c. Content of Computing principles (Component 01)

This component will introduce learners to the internal workings of the Central Processing Unit (CPU), the exchange of data and will also look at software development, data types and legal and ethical issues. It is expected that learners will draw on this underpinning content when studying computational thinking and developing programming techniques.

Learners will be expected to apply the criteria, in the tables below, in different contexts including current and future uses of the technologies.

2

1.1 The characteristics of contemporary processors, input, output and storage devices	
Components of a computer and their uses	
1.1.1 Structure and function of the processor	(a) The Arithmetic and Logic Unit; ALU, Control Unit and Registers (Program Counter; PC, Accumulator; ACC, Memory Address Register; MAR, Memory Data Register; MDR, Current Instruction Register; CIR). Buses: data, address and control: how this relates to assembly language programs. (b) The fetch-decode-execute cycle, including its effect on registers. (c) The factors affecting the performance of the CPU, clock speed, number of cores, cache. (d) Von Neumann, Harvard and contemporary processor architecture.
1.1.2 Types of processor	(a) The differences between and uses of CISC and RISC processors. (b) Multicore and Parallel systems.
1.1.3 Input, output and storage	(a) How different input, output and storage devices can be applied to the solution of different problems. (b) The uses of magnetic, flash and optical storage devices. (c) RAM and ROM. (d) Virtual storage.

1.2 Software and software development	
Types of software and the different methodologies used to develop software	
1.2.1 Operating Systems	(a) The need for, function and purpose of operating systems. (b) Memory Management (paging, segmentation and virtual memory). (c) Interrupts, the role of interrupts and Interrupt Service Routines (ISR), role within the Fetch-Decode-Execute Cycle. (d) Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time. (e) Distributed, embedded, multi-tasking, multi-user and real time operating systems. (f) BIOS. (g) Device drivers. (h) Virtual machines, any instance where software is used to take on the function of a machine including executing intermediate code or running an operating system within another.
1.2.2 Applications generation	(a) The nature of applications, justifying suitable applications for a specific purpose. (b) Utilities. (c) Open source vs closed source. (d) Translators: Interpreters, compilers and assemblers.
1.2.3 Introduction to programming	(a) Procedural programming language techniques: • program flow • variables and constants • procedures and functions • arithmetic, Boolean and assignment operators • string handling • file handling. (b) Assembly language (including following and writing simple programs with Little Man Computer). See appendix 5d.

1.3 Exchanging data	
How data is exchanged between different systems	
1.3.1 Databases	(a) Relational database, flat file, primary key, foreign key, secondary key, entity relationship modelling. See appendix 5d and 5e. (b) Methods of capturing, selecting, managing and exchanging data.
1.3.2 Networks	(a) Characteristics of networks and the importance of protocols and standards. (b) Internet structure: • The TCP/IP stack. • DNS. • Protocol layering. • LANs and WANs. • Packet and circuit switching. (c) Client-server and peer to peer.
1.3.3 Web Technologies	(a) HTML, CSS and JavaScript. See appendix 5d. (b) Lossy v lossless compression.

1.4 Data types, data structures and algorithms	
How data is represented and stored within different structures. Different algorithms that can be applied to these structures	
1.4.1 Data Types	(a) Primitive data types, integer, real/floating point, character, string and Boolean. (b) Represent positive integers in binary. (c) Use of sign and magnitude and two's complement to represent negative numbers in binary. (d) Addition and subtraction of binary integers. (e) Represent positive integers in hexadecimal. (f) Convert positive integers between binary hexadecimal and denary. (g) Positive and negative real numbers using normalised floating point representation. (h) How character sets (ASCII and UNICODE) are used to represent text.
1.4.2 Data Structures	(a) Arrays (of up to 3 dimensions), records, lists, tuples. (b) The properties of stacks and queues.
1.4.3 Boolean Algebra	(a) Define problems using Boolean logic. See appendix 5d. (b) Manipulate Boolean expressions, including the use of Karnaugh maps to simplify Boolean expressions. (c) Use logic gate diagrams and truth tables.

1.5 Legal, moral, ethical and cultural issues	
The individual moral, social, ethical and cultural opportunities and risks of digital technology. Legislation surrounding the use of computers and ethical issues that can or may in the future arise from the use of computers	
1.5.1 Computing related legislation	(a) The Data Protection Act 1998. (b) The Computer Misuse Act 1990. (c) The Copyright Design and Patents Act 1988. (d) The Regulation of Investigatory Powers Act 2000.
1.5.2 Ethical, moral and cultural issues	(a) The individual moral, social, ethical and cultural opportunities and risks of digital technology: • Computers in the workforce. • Automated decision making. • Artificial intelligence. • Environmental effects. • Censorship and the Internet. • Monitor behaviour. • Analyse personal information. • Piracy and offensive communications. • Layout, colour paradigms and character sets.

2c. Content of Algorithms and problem solving (Component 02)

This component will incorporate and build on the knowledge and understanding gained in the Computing principles component (01). In addition, learners should:

- understand what is meant by computational thinking
- understand the benefits of applying computational thinking to solving problems
- be able to use algorithms to describe problems.

2

2.1 Elements of computational thinking	
Understand what is meant by computational thinking	
2.1.1 Thinking abstractly	(a) The nature of abstraction. (b) The need for abstraction. (c) The differences between an abstraction and reality. (d) Devise an abstract model for a variety of situations.
2.1.2 Thinking ahead	(a) Identify the inputs and outputs for a given situation. (b) Determine the preconditions for devising a solution to a problem. (c) The need for reusable program components.
2.1.3 Thinking procedurally	(a) Identify the components of a problem. (b) Identify the components of a solution to a problem. (c) Determine the order of the steps needed to solve a problem. (d) Identify sub-procedures necessary to solve a problem.
2.1.4 Thinking logically	(a) Identify the points in a solution where a decision has to be taken. (b) Determine the logical conditions that affect the outcome of a decision. (c) Determine how decisions affect flow through a program.

2.2 Problem solving and programming	
How computers can be used to solve problems and programs can be written to solve them *(Learners will benefit from being able to program in a procedural/imperative language.)*	
2.2.1 Programming techniques	(a) Programming constructs: sequence, iteration, branching. (b) Global and local variables. (c) Modularity, functions and procedures, parameter passing by value and reference. (d) Use of an IDE to develop/debug a program.
2.2.2 Software Development	(a) Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development. (b) The relative merits and drawbacks of different methodologies and when they might be used. (c) Writing and following algorithms. (d) Different test strategies, including black and white box testing and alpha and beta testing. (e) Test programs that solve problems using suitable test data and end user feedback, justify a test strategy for a given situation.

2.3 Algorithms	
The use of algorithms to describe problems and standard algorithms	
2.3.1 Algorithms	(a) Analysis and design of algorithms for a given situation. (b) Standard algorithms (bubble sort, insertion sort, binary search and linear search). (c) Implement bubble sort, insertion sort. (d) Implement binary and linear search. (e) Representing, adding data to and removing data from queues and stacks. (f) Compare the suitability of different algorithms for a given task and data set.

2d. Prior learning and progression

There are no prior qualification requirements for this specification.

Learners in England who are beginning an AS level course are likely to have followed a Key Stage 4 programme of study. This course will enable learners to progress to higher study or to progress directly to employment.

The qualification is suitable for learners intending to pursue any career in which an understanding of technology is needed. The qualification is also suitable for any further study as part of a course of general education.

The qualification will provide learners with a range of transferable skills which will facilitate personal growth and foster cross curriculum links in areas such as maths, science and design and technology. Computer Science is a very creative subject and skills such as problem solving and analytical thinking will all be refined and explored as learners progress through the learning and assessment programme.

There are a number of Computer Science qualifications at OCR. Find out more at **www.ocr.org.uk**

3 Assessment of OCR AS Level in Computer Science

3a. Forms of assessment

The AS Level in Computer Science is a linear qualification with 100% terminal external assessment.

This qualification consists of two examined components (01 and 02), both examinations are of 1 hour and 15 minutes duration, each with a 50% weighting.

Computing principles (Component 01)

Learners answer **all** the questions. There will be a mix of questions including short answer, longer answer and some higher tariff questions that will test the quality of extended responses. Marks for these responses are integrated into the marking criteria.

The whole of the Computing principles content will be covered over the life of the specification.

Algorithms and problem solving (Component 02)

Learners answer **all** the questions. There will be a mix of questions including short answer, longer answer and some higher tariff questions that will test the quality of extended responses. Marks for these responses are integrated into the marking criteria.

The whole of the Algorithms and problem solving content will be covered over the life of the specification.

3b. Assessment objectives (AO)

There are three assessment objectives for OCR's AS Level in Computer Science and these are detailed in the table below.

Learners are expected to demonstrate their ability to:

	Assessment Objective
AO1	Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation.
AO2	Apply knowledge and understanding of the principles and concepts of computer science, including to analyse problems in computational terms.
AO3	Design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions.

AO weightings in AS Level in Computer Science

The relationship between the assessment objectives and the components are shown in the following table:

Component	% of AS Level Computer Science (H046)			
	AO1	AO2	AO3	Total
Computing principles (H046/01)	19*	16*	15*	50
Algorithms and problem solving (H046/02)	17*	19*	14*	50
Total	**36***	**35***	**29***	**100**

* values rounded to nearest whole %.

3c. Assessment availability

There will be one examination series available each year in May/June to **all** learners.

This specification will be certificated from the June 2016 examination series onwards.

3d. Retaking the qualification

Learners can retake the qualification as many times as they wish.

They retake both components of the qualification.

3e. Assessment of extended responses

The assessment materials for this qualification provide learners with the opportunity to demonstrate their ability to construct and develop a sustained line of reasoning and marks for extended responses are integrated into the marking criteria.

3f. Synoptic assessment

Synoptic assessment draws together the knowledge, understanding and skills learnt in different aspects of the AS Level in Computer Science course.

The emphasis of synoptic assessment is to encourage the understanding of Computer Science as a discipline.

It is envisaged that the Computing principles (component 01) would be taught first but there are elements of the Algorithms and problem solving (component 02) that should also be considered. Computing principles is taught in order that learners are fully aware of the way problem solving through computational thinking is achieved.

The Algorithms and problem solving (component 02) builds on the knowledge and understanding gained in the Computing principles (component 01). Algorithms and problem solving will contain synoptic assessment.

3g. Calculating qualification results

A learner's overall qualification grade for an AS Level in Computer Science will be calculated by adding together their marks from the two components taken to give their total weighted mark.

This mark will then be compared to the qualification level grade boundaries for the qualification for the relevant exam series to determine the learner's overall qualification grade.

4 Admin: what you need to know

The information in this section is designed to give an overview of the processes involved in administering this qualification so that you can speak to your exams officer. All of the following processes require you to submit something to OCR by a specific deadline. More information about these processes, together with the deadlines, can be found in the OCR *Admin Guide and Entry Codes: 14–19 Qualifications*, which can be downloaded from the OCR website: www.ocr.org.uk

4a. Pre-assessment

Estimated entries

Estimated entries are your best projection of the number of learners who will be entered for a qualification in a particular series.

Estimated entries should be submitted to OCR by the specified deadline. They are free and do not commit your centre in any way.

Final entries

Final entries provide OCR with detailed data for each learner, showing each assessment to be taken. It is essential that you use the correct entry and option code.

Final entries must be submitted to OCR by the published deadlines or late entry fees will apply.

All learners taking AS Level in Computer Science must be entered for H046.

Entry option		Components		
Entry code	Title	Code	Title	Assessment type
H046	Computer Science	01	Computing principles	External Assessment
		02	Algorithms and problem solving	External Assessment

4b. Accessibility and special consideration

Reasonable adjustments and access arrangements allow learners with special educational needs, disabilities or temporary injuries to access the assessment and show what they know and can do, without changing the demands of the assessment. Applications for these should be made before the examination series. Detailed information about eligibility for access arrangements can be found in the JCQ *Access Arrangements and Reasonable Adjustments*.

Special consideration is a post-assessment adjustment to marks or grades to reflect temporary injury, illness or other indisposition at the time the assessment was taken.

Detailed information about eligibility for special consideration can be found in the JCQ *A guide to the special consideration process*.

4c. External assessment arrangements

Regulations governing examination arrangements are contained in the JCQ *Instructions for conducting examinations*.

Learners are permitted to use a scientific or graphical calculator for both components. Calculators are subject to the rules in the document *Instructions for Conducting Examinations* published annually by JCQ (www.jcq.org.uk).

Head of Centre Annual Declaration

The Head of Centre is required to provide a declaration to the JCQ as part of the annual NCN update, conducted in the autumn term, to confirm that the centre is meeting all of the requirements detailed in the specification.

Any failure by a centre to provide the Head of Centre Annual Declaration will result in your centre status being suspended and could lead to the withdrawal of our approval for you to operate as a centre.

4d. Results and certificates

Grade scale

AS level qualifications are graded on the scale: A, B, C, D, E, where A is the highest. Learners who fail to reach the minimum standard for E will be Unclassified (U).

Only subjects in which grades A to E are attained will be recorded on certificates.

Results

Results are released to centres and learners for information and allow any queries to be resolved **before** certificates are issued.

Centres will have access to the following results information for each learner:

- the grade for the qualification
- the raw mark for each component
- the total weighted mark for the qualification.

The following supporting information will be available:

- raw mark grade boundaries for each component
- weighted mark grade boundaries for the qualification.

Until certificates are issued, results are deemed to be provisional and may be subject to amendment. A learner's final results will be recorded on an OCR certificate.

The qualification title will be shown on the certificate as 'OCR Level 3 Advanced Subsidiary GCE in Computer Science'.

4e. Post-results services

A number of post-results services are available:

Enquiries about results – If you are not happy with the outcome of a learner's results, centres may submit an enquiry about results.

Missing and incomplete results – This service should be used if an individual subject result for a learner is missing, or the learner has been omitted entirely from the results supplied.

Access to scripts – Centres can request access to marked scripts.

4f. Malpractice

Any breach of the regulations for the conduct of examinations and coursework may constitute malpractice (which includes maladministration) and must be reported to OCR as soon as it is detected.

Detailed information on malpractice can be found in the JCQ publication *Suspected Malpractice in Examinations and Assessments: Policies and Procedures*.

5 Appendices

5a. Overlap with other qualifications

The knowledge, understanding and skills that are developed throughout this qualification are distinct and have very little overlap with other qualifications.

This overlap may occur only at level 5 in the hardware and software elements of ICT Cambridge Technicals and GCE Applied ICT.

5b. Avoidance of bias

The A level qualification and subject criteria have been reviewed in order to identify any feature which could disadvantage candidates who share a protected Characteristic as defined by the Equality Act 2010. All reasonable steps have been taken to minimise any such disadvantage.

5c. Mathematical skills

Computer Science uses mathematics to express its computational laws and processes.

All AS level and A level Computer Science qualifications must contain a minimum of 10% mathematical skills. Candidates may be asked to demonstrate their knowledge, understanding and skills of computational processes and problem solving in both theoretical and practical ways. The following list of topics will be counted as Level 2 (or higher) mathematics.

Topic:

- Boolean algebra
- Number representation and bases

Whilst the concept for each topic is Level 2 (though it may not appear in GCSE mathematics specifications) candidates will, however be expected to apply the skills in a Level 3 context.

5d. Languages and Boolean logic guide for use in external assessments

The tables below show languages and logic that will be used in the external assessments and indicates the limits and scope of each.

Centres are free to go beyond these parameters.

Pseudocode

The following guide shows the format pseudocode will appear in the examined units. It is provided to allow you to give learners familiarity before the exam. Learners are not expected to memorise the syntax of this pseudocode and when asked may provide answers in any style of pseudocode they choose providing its meaning could be reasonably inferred by a competent programmer.

Variables
Variables are assigned using the = operator.

```
x=3
name="Bob"
```

A variable is declared the first time a value is assigned. It assumes the data type of the value it is given.

Variables declared inside a function or procedure are local to that subroutine.

Variables in the main program can be made global with the keyword `global`.

```
global userid = 123
```

Casting
Variables can be typecast using the `int str` and `float` functions.

`str(3)` returns "3"
`int ("3")` returns 3
`float ("3.14")` returns `3.14`

Outputting to Screen
`print(`*string*`)`

Example

```
print("hello")
```

Taking Input from User
variable`=input(`*prompt to user*`)`

Example

```
name=input("Please enter your name")
```

Iteration – Count Controlled

```
for i=0 to 7
    print("Hello")
next i
```

Will print hello 8 times (0–7 inclusive).

Iteration – Condition Controlled

```
while answer!="computer"
    answer=input("What is the password?")
endwhile
```

```
do
    answer=input("What is the password?")
until answer=="computer"
```

Logical Operators

```
AND OR NOT
```

```
e.g.
while x<=5 AND flag==false
```

Comparison Operators

==	Equal to
!=	Not equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to

Arithmetic Operators

+	Addition e.g. x=6+5 gives 11
–	Subtraction e.g. x=6–5 gives 1
*	Multiplication e.g. x=12*2 gives 24
/	Division e.g. x=12/2 gives 6
MOD	Modulus e.g. 12MOD5 gives 2
DIV	Quotient e.g. 17DIV5 gives 3
^	Exponentiation e.g. 3^4 gives 81

Selection

Selection will be carried out with if/else and switch/case

if/else

```
if entry==”a” then
    print(“You selected A”)
elseif entry==”b” then
    print(“You selected B”)
else
    print(“Unrecognised selection”)
endif
```

switch/case

```
switch entry:
     case “A”:
        print(“You selected A”)
     case “B”:1
        print(“You selected B”)
     default:
        print(“Unrecognised selection”)

endswitch
```

String Handling

To get the length of a string:
stringname.length

To get a substring:

```
stringname.subString(startingPosition, numberOfCharacters)
```

NB The string will start with the 0^{th} character.

Example

```
someText=”Computer Science”

print(someText.length)
print(someText.substring(3,3))
```

Will display

```
16
put
```

Subroutines

```
function triple(number)
    return number*3
endfunction
```

Called from main program

```
y=triple(7)
```

```
procedure greeting(name)
    print("hello"+name)
endprocedure
```

Called from main program

```
greeting("Hamish")
```

Unless stated values passed to subroutines can be assumed to be passed by value.
If this is relevant to the question byVal and byRef will be used. In the case below x is passed by value and y is passed by reference.

```
procedure foobar(x:byVal, y:byRef)
    …
    …
endprocedure
```

5

Arrays

Arrays will be 0 based and declared with the keyword *array*.

```
array names[5]
names[0]="Ahmad"
names[1]="Ben"
names[2]="Catherine"
names[3]="Dana"
names[4]="Elijah"

print(names[3])
```

Example of 2D array:

```
Array board[8,8]
board[0,0]="rook"
```

Reading to and Writing from Files

To open a file to read from openRead is used and readLine to return a line of text from the file.

The following program makes x the first line of sample.txt

```
myFile = openRead("sample.txt")
x = myFile.readLine()
myFile.close()
```

endOfFile() is used to determine the end of the file. The following program will print out the contents of sample.txt

```
myFile = openRead("sample.txt")
while NOT myFile.endOfFile()
    print(myFile.readLine())
endwhile
myFile.close()
```

To open a file to write to openWrite is used and writeLine to add a line of text to the file. In the program below hello world is made the contents of sample.txt (any previous contents are overwritten).

```
myFile = openWrite("sample.txt")
myFile.writeLine("Hello World")
myFile.close()
```


Comments

Comments are denoted by //

```
print("Hello World") //This is a comment
```

HTML

Learners are expected to have an awareness of the following tags. Any other tags used will be introduced in the question.

<html>
<link> to link to a CSS file
<head>

<title>

<body>

<h1> <h2> <h3>

<img> including the src, alt, height and width attributes.

<a> including the href attribute.

<form>

<input> where the input is a textbox (i.e. has the attribute type="text" and another attribute name to identify it) or a submit button (i.e. has the attribute type="submit")

<p>

<script>

Any other elements used will be explained in the question.

CSS

Learners are expected to be able to use CSS directly inside elements using the style attribute

```
<h1 style="color:blue;">
```

and external style sheets. In the style sheets they should be able to use CSS to define the styling of elements:

```
h1{
    color:blue;
}
```

classes

```
.infoBox{
    background-color: green;
}
```

and Identifiers

```
#menu{
    background-color: #A2441B
}
```

They are expected to be familiar with the following properties.

`background-color`
`border-color`
`border-style`
`border-width`
`color` *with named and hex colours*
`font-family`
`font-size`
`height`
`width`

Any other properties used will be explained in the question.

5

JavaScript

Learners are expected to be able to follow and write basic JavaScript code. It is hoped they will get practical experience of JavaScript in their study of the course. They will not be expected to commit exact details of syntax to memory. Questions in the exam will not penalise learners for minor inaccuracies in syntax. Learners *will* be expected to be familiar with the JavaScript equivalents of the structures listed in the pseudocode section (with the exception of input and output (see below). They will not be expected to use JavaScript for Object Oriented programming or file handling. Questions will not be asked in JavaScript where something is passed to a subroutine by value or reference is relevant.

Input

Input will be taken in by reading values from a form. *NB learners will not be expected to memorise the method for doing this as focus will be on what they do with that input once it is received.*

Output

By changing the contents of an HTML element

```
chosenElement = document.getElementById("example");
chosenElement.innerHTML = "Hello World";
```

By writing directly to the document

```
document.write("Hello World");
```

By using an alert box

```
alert("Hello World");
```

Should any other aspects of JavaScript be used they will be introduced and explained in the question.

Little Man Computer Instruction Set

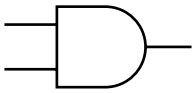
In questions mnemonics will always be given according to the left hand column below. Different implementations of LMC have slight variations in mnemonics used and to take this into account the alternative mnemonics in the right hand column will be accepted in learners' answers.

Mnemonic	Instruction	Alternative mnemonics accepted
ADD	Add	
SUB	Subtract	
STA	Store	STO
LDA	Load	LOAD
BRA	Branch always	BR
BRZ	Branch if zero	BZ
BRP	Branch if positive	BP
INP	Input	IN, INPUT
OUT	Output	
HLT	End program	COB, END
DAT	Data location	

Boolean Algebra

When Boolean algebra is used in questions the notation described below will be used. Other forms of notation exist, and so below is a list of accepted notation we will accept from learners.

Conjunction



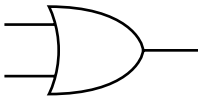
Notation used:
$\wedge$ e.g. A $\wedge$ B

A	B	A $\wedge$ B
T	T	T
T	F	F
F	T	F
F	F	F

Alternatives accepted:

AND e.g. A AND B
e.g. A.B

Disjunction



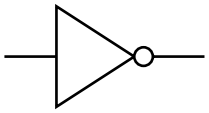
Notation used:
$\vee$ e.g. A $\vee$ B

A	B	A $\vee$ B
T	T	T
T	F	T
F	T	T
F	F	F

Alternatives accepted:

OR e.g. A OR B
\+ e.g. A+B

Negation

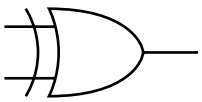


Notation used:
¬ e.g. ¬A

A	¬A
T	F
F	T

Alternatives accepted:
bar e.g. $\overline{A}$
~ e.g. ~A
NOT e.g. NOT A

Exclusive Disjunction



Notation used:
$\veebar$ e.g. A $\veebar$ B

A	B	A $\veebar$ B
T	T	F
T	F	T
F	T	T
F	F	F

Alternatives accepted:

XOR e.g. A XOR B
$\oplus$ e.g. A $\oplus$B

Equivalence / Iff

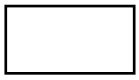
Notation used:
$\equiv$ e.g. $(A \barwedge B) \equiv \neg(\neg A \barvee \neg B)$

Alternative accepted:
$\leftrightarrow$

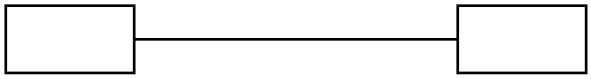
5e. Entity relationship diagrams

The following symbols are used for entities and their relationships.

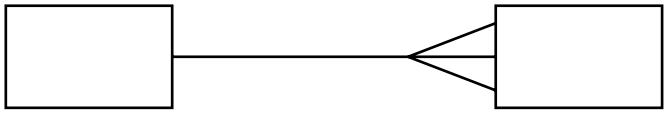
Entity



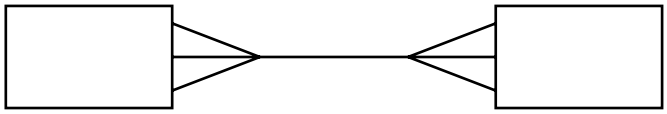
One-To-One relationship



One-To-Many relationship



Many-To-Many relationship



Your checklist

Our aim is to provide you with all the information and support you need to deliver our specifications.

- [ ] Bookmark **ocr.org.uk/alevelcomputerscience** for all the latest resources, information and news on AS and A Level Computer Science
- [ ] Be among the first to hear about support materials and resources as they become available – register for Computer Science updates at **ocr.org.uk/updates**
- [ ] Find out about our professional development at **cpdhub.ocr.org.uk**
- [ ] View our range of skills guides for use across subjects and qualifications at **ocr.org.uk/skillsguides**
- [ ] Discover our new online past paper service at **ocr.org.uk/examcreator**
- [ ] Learn more about Active Results at **ocr.org.uk/activeresults**
- [ ] Join our Computer Science social network community for teachers at **social.ocr.org.uk**

Download high-quality, exciting and innovative AS and A Level Computer Science resources from ocr.org.uk/alevelcomputerscience

Free resources and support for our AS Level Computer Science qualification, developed through collaboration between our Computer Science Subject Specialists, teachers and other subject experts, are available from our website. You can also contact our Computer Science Subject Specialists for specialist advice, guidance and support, giving you individual service and assistance whenever you need it.

Meet the team at ocr.org.uk/computerscienceteam and contact them at:
01223 553998
computerscience@ocr.org.uk
@OCR_ict

To stay up to date with all the relevant news about our qualifications, register for email updates at ocr.org.uk/updates

Computer Science community

The social network is a free platform where teachers can engage with each other – and with us – to find and offer guidance, discover and share ideas, best practice and a range of Computer Science support materials. To sign up, go to social.ocr.org.uk

follow us on

facebook.com/ocrexams
linkedin.com/company/ocr
@OCR_ict
youtube.com/ocrexams

PART OF THE CAMBRIDGE ASSESSMENT GROUP